

MODULE FOR THE INTERCONNECTIVITY OF INDEPENDENT SOFTWAREAPPLICATIONSCROSS-REFERENCES TO RELATED APPLICATIONS

[0001] This application claims the benefit of Provisional Patent Application Serial No. 60/194,377, filed on April 4, 2000, entitled "Module For The Interconnectivity Of Independent Software Applications," which is incorporated herein by reference.

BACKGROUND OF THE INVENTIONField of the Invention

[0002] The present invention relates to data processing systems, and specifically to data processing systems that interface between multiple independent software applications. In particular, the invention is a module for interfacing between at least a payroll application, an accounts payable application and a time and expense application.

Description of Related Art

[0003] Numerous independent programs exist for keeping track of employee time slips, payroll, accounts payable, projects, and billing. Unfortunately, these independent programs are not designed to communicate with one another and, as a result, time consuming duplicate entries of data is often required. The duplicative entry of data and the potential for erroneous data entry is unacceptable for both large companies employing a large number of full-time employees and for temporary service agencies employing many part time employees. Therefore, there exists a need in the art for a program module that interfaces between at least a payroll application, an accounts payable application, and a time and expense application.

SUMMARY OF THE INVENTION

[0004] The present invention discloses an interconnectivity program module that provides for interconnectivity and interfacing of independent application programs in the context of a data processing system. Specifically, the interconnectivity program module provides a time and expenses database and interface capability between a payroll application, an accounts payable application, and a projects/billing application.

[0005] Once data is entered into the time and expenses database of the interconnectivity program module, the data is available to a payroll processor, an accounts payable processor and a projects/billing processor. These processors run independent payroll, accounts payable and projects/billing software applications, respectively, without compromising the integrity and accuracy of the data. As a result, duplicate entries to each processor are eliminated.

[0006] The simultaneous entry of data to several processing programs in this way ensures consistent and correct data entry between applications. The present invention also allows for the prompt and accurate disbursement of payments to employees and vendors, as well as the proper and accurate billing of customers.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] Fig. 1 is a diagram of an enterprise-wide data processing system utilizing an interconnectivity program module of the present invention.

[0008] Fig. 2 shows the steps used to input time and expense data into a time and expenses database of the present invention.

[0009] Fig. 3 shows the steps used to interconnect time and expense data from the time and expenses database of Fig. 2 to a payroll processing system.

[0010] Fig. 4 shows the steps used to interconnect time and expense data from the time and expenses database of Fig. 2 to an accounts payable system.

[0011] Fig. 5 shows the interconnectivity relationship between the interconnectivity program module, the payroll processor, the accounts payable processor and the projects/billing processor.

[0012] Fig. 6 shows the steps used to interconnect payroll data and expense data from the payroll processor and the accounts payable processor, respectively, to a projects/billing system.

DESCRIPTION OF THE PREFERRED EMBODIMENT

[0013] The present invention discloses a system and method for interconnecting independent software applications. The invention is an interconnectivity software program that provides the interfacing and interconnecting of data between one or more independent applications.

[0014] Fig. 1 shows the architecture of a system for interconnecting various independent software applications according to the present invention. In the preferred embodiment, an interconnectivity program module 100 interfaces with a Human Resources Management System (HRMS) server 120 and a financials server 110, as shown in Fig. 1. The interconnectivity program module 100 provides communications to each of the servers by way of an HRMS interface 125 and a financials interface 115.

[0015] The purpose of each server is to run one or more independent software applications in order to process data for specific tasks. In the preferred embodiment, the financials server 110 includes an accounts payable system module 130 and a projects system module 140, both of which run independent applications, such as projects, billing, and accounts payable programs. The accounts payable system module 130 and the projects system module 140 process data provided by the financials server 110 through an accounts payable interface 135 and a projects interface 145, respectively. The HRMS server 120 also runs independent applications, such as a payroll import program 170 and a payroll-processing program 180. Thus, the system can run applications that process data for time cards, payroll, project, billing, costing and accounts payable functions. Many other independent software applications can be provided in this system, thus the present invention is not limited to the applications exemplified in Fig. 1.

[0016] Both the financials server 110 and the HRMS server 120 contain a database that is specific to the information that the particular server processes. As can be seen in Fig. 1, the HRMS server 120 uses an HRMS database 121 and the financials server 110 uses a financials database 111. As will be described below, these databases contain time and expense data used for processing according to specific requirements.

[0017] A data entry operator 150 enters data relating to various parameters, such as employee time card data and expenses, into the interconnectivity program module 100 by way of a series of rapid entry panels 160. The data is then sent through a data entry interface 155 to the interconnectivity program module 100, where it is stored in a time and expenses database 105. The data is kept in the time and expenses database 105 until needed for processing by the HRMS server 120 and/or the financials server 110.

[0018] Fig. 2 shows the how data is inputted to the interconnectivity program module 100 by way of the data entry interface 155 (Fig. 1). In some work environments, employees use conventional time cards to log time worked. In steps 200b and 200c, the employee enters his or her time into the system either by manually filling in the time worked on a paper time card or by punching the card with a mechanical clock. In step 201, the data entry operator 150 (Fig. 1) takes a large number of these physical time cards, sorts them by branch and type, and reviews the time card feeds. Then, in step 202, the data entry operator 150 creates control batches for entry into the interconnectivity program module 100. In step 203a, the data entry operator 150 can use the rapid entry panels 160 (Fig. 1) to manually enter both time card details and expense data. The rapid entry panels 160 provide the data entry operator 150 with a variety of templates to use for data entry. Because the interconnectivity program module 100 (Fig. 1) is capable of receiving input in a multiplicity of time and expense program

PA1601US

formats, the rapid entry panels 160 allow the data entry operator 150 to format the outgoing data in a number of different formats that are compatible with the interconnectivity program module 100.

[0019] In other work environments, time may be entered electronically, such as through a computer or use of a time card with a magnetic stripe for swiping through a time card reader. In this case, the employee enters his or her time electronically in step 200a. Thus, the time card details and expense data are directly entered into the data processing system in step 203b without manual intervention from the data entry operator 150.

[0020] In step 204, real-time verification of the data is performed. In the preferred embodiment, the data is verified using a human resources database 210 and a projects costing database 220. The data is compared to data in the human resources database 210 and the projects costing database 220 and may be verified and edited at this step. Such information includes verifying whether the employee is in active status, verifying an employee's pay rate, verifying vacation and holiday hours, applying special overtime rules, verifying dates of employment with dates of pay, verifying employee billing rates, etc.

[0021] Step 205 determines whether or not the time and expense data entries have passed high-level audits. In the preferred embodiment, high level audits may include checking whether the hourly totals add up to an eight hour work day, whether the start and stop times are within a predefined period, and whether the employee is properly assigned to the billed job. If a particular data entry does not pass the high level audits, step 206 preferably returns the data for online correction by the data entry operator 150. The data is then sent through the high level audit in step 205 for re-evaluation. Once the data passes the high level audits, the data entry operator 150 runs the data entry interface 155 in step 207.

[0022] The data entry interface 155 performs one or more low-level data audits in step 208. Low-level data audits are specific to a job and/or customer. In the preferred embodiment, this may include audits as to mandatory lunch hours, applicable hourly pay rates for overtime, required workdays, or any other auditable criteria. If the data fails the low-level audits in step 208, step 209 allows the data entry operator 150 to review and correct the data online. The data is then sent through the low level audits of step 208 for re-evaluation. Once the data passes the low-level audits of step 208, the data entry interface 155 sends the data to the time and expenses database 105 for storage and future retrieval. The time and expense data is now available for use by the financials server 110 and the HRMS server 120 (Fig. 1).

[0023] Fig. 3 shows how the interconnectivity program module 100 (Fig. 1) uses the HRMS interface 125 (Fig. 1) to provide an interface between data in the time and expenses database 105 (Fig. 1) and applications running on the HRMS server 120 (Fig. 1). In step 300, the data entry operator 150 runs the HRMS interface 125 on payroll data stored in and retrieved from the time and expenses database 105. In step 301, the HRMS interface 125 creates files from the payroll data that is compatible with the payroll import program 170 running on the HRMS server 120 (Fig. 1) and sends the converted data to the payroll import program 170. An example of the payroll import program 170 is Pay Sheets, manufactured and distributed by PeopleSoft, Inc. of Pleasanton, California. In step 302, the payroll import program 170 converts the files into a format that is compatible with the payroll-processing program 180, also running on the HRMS server 120 (Fig. 1), and sends the files to the payroll-processing program. In step 303, the payroll-processing program 180 confirms the payroll data and generates a check 370 or a direct deposit 380 for an employee.

[0024] In step 304, after payroll confirmation in step 303, the data processed by the payroll-processing program 180 is sent as payroll data 390 to the projects system module 140 by way of the projects interface 145 (see discussion below with respect to Figs. 5 and 6).

[0025] Fig. 4 shows how the interconnectivity program module 100 (Fig. 1) uses the financials interface 115 (Fig. 1) to provide an interface between data in the time and expenses database 105 (Fig. 1) and applications running on the accounts payable system module 130 (Fig. 1) of the financials server 110 (Fig. 1) by way of the accounts payable interface 135 (Fig. 1). In step 400 of Fig. 4, the data entry operator 150 runs the financials interface 115 on expense data stored in and retrieved from the time and expenses database 105. The financials interface sends expense data to the financials server 110. In step 401, the accounts payable interface 135 runs the data on the accounts payable system module 130. The accounts payable system module 130 may include an accounts payable program 460, which also resides on the financials server 110. In step 402, the data is processed by the accounts payable program 460, which generates a payment 470 for the appropriate vendor to whom the payment is owed. In step 403, the data processed by the accounts payable program 460 is sent as expense data 490 to the projects system module 140 (Fig. 1) by way of the projects interface 145 (see discussion below with respect to Figs. 5 and 6).

[0026] Fig. 5 shows the data processing system of Fig. 1 and how the interconnectivity program module 100 provides the interfacing of the payroll data 390 and the expense data 490 with the projects interface 145 as described above in reference to Figs. 3 and 4. Using the present invention, the payroll data 390 and the expense data 490 is shared with the projects system module 140 after processing by the independent applications running on the financials server 110 and the HRMS server 120.

[0027] Fig. 6 shows how data in the time and expenses database 105, the payroll data 390 and the expense data 490 interface with the projects system module 140 (Fig. 1). As discussed above, in step 304 (Fig. 3) and 403 (Fig. 4), the payroll data 390 and the expense data 490 processed by the payroll-processing program 180 (Fig. 1) and the accounts payable program 460 (Fig. 4), respectively, are sent to the projects system module 140 by way of the projects interface 145 (Fig. 1). At step 600, the data entry operator 150 runs the projects interface 145, which incorporates all payroll, labor, expenses and vendor costs into the data processing system. In step 601, the projects interface 145 sends the data to a billing program 650. In step 602, the billing program 650 picks up the appropriate transactions entered in step 600 for standard billing processing. The billing program 650 then generates an invoice 660 to customers. In step 603, the data entered from step 600 is made available to generate a report 670. There are a number of reports and analyses that can be generated, including reports that detail expenses, efficiency, etc.

[0028] Using the present invention as described above, single entries of time and expense data are made through the interconnectivity program module 100. The data is stored and used by any number of independent software applications running on the HRMS server 120 and the financials server 110, eliminating duplicate entries of the data. The single entry of data to several processing programs in this way ensures consistent and correct data entry between applications. It also ensures that employees and vendors are paid the correct amount, and that customers are billed properly and accurately.

[0029] As preferred embodiments of the present invention are described above with reference to the aforementioned drawings, various modifications or adaptations of the methods and or specific structures described may become apparent to those skilled in the art. All such modifications, adaptations,

PA1601US

or variations that rely upon the teachings of the present invention, and through which these teachings have advanced the art, are considered to be within the spirit and scope of the present invention. Hence, these descriptions and drawings are not be considered in a limiting sense as it is understood that the present invention is in no way limited to the embodiments illustrated.